HTML (Part I): The Elements

*By Kevin Savetz**

So you're ready to create your own World Wide Web page—a masterpiece of audio and visual delights to market your product and win new customers. The only thing between you and your own storefront on the Internet is something called HTML, the Hypertext Markup Language.

HTML is simply a set of commands for placing elements on a "page," or document, the unit of information on the Web. Despite what the ads for big books and expensive online storefronts will tell you, learning HTML is relatively painless, if you are willing to roll up your sleeves and experiment a little.

This is the first of three articles on HTML. The next article will introduce the advanced features of HTML. The final article looks at style issues. The author, Kevin Savetz, is taking time off from his Help Desk column to write the series. You can visit his personal home page at* ***http://www.northcoast.com/~savetz/savetz.html.*

What You Need

To create HTML files, all you need is a basic text editor. Any word processor that can save files as ASCII text will do. Other programs can simplify the creation of HTML by remembering the more complex commands for you. For now, though, just load your word processor and experiment. You don't even need to be running a World Wide Web server to look at your creation. Most Web browsers (including Mosaic and Netscape) let you read an HTML file stored on your hard disk. With a Web browser, you can create pages, then test and refine them, without even going on the Web.

With HTML you specify how the elements of a "page"—headers, lists, text, and graphics, for example—will appear when viewed by someone's Web browser ("client" software). All browsers work by downloading text files marked up with HTML codes, then displaying formatted pages on the local computer. Different browsers display HTML code in slightly different ways, so it's a good idea to test pages using as many browsers as possible. The page itself is situated on the World Wide Web "server" maintained by you or your access provider. (See Sections 960 and 1050 on Web client and server software, respectively.)

Note that Web browsers often let users set the default type size, style, and color. See the figure for a representation of the finished page the user sees.

Basic Text

HTML uses formatting codes, called *tags*, to define the appearance of document elements. Tags always go within angle brackets: <>. For instance, to create a line of large headline text (a "first-level header"), we use the <h1> tag to turn on the tag:

```
<h1>Welcome to Arlo's Rare Pet Store</h1>
```

A forward slash (/) turns the tag off, so </h1> turns off the headline text. HTML provides us with a variety of headline sizes, from <h1> (really big) to <h6> (really small). Text that isn't tagged is treated as plain old body text, the on-screen equivalent of the text in this paragraph.

```
<h1>Welcome to Arlo's Rare Pet Store</h1>
Greetings from Arlo's Rare pets, your one-
stop shop for unusual pets and everything
they need.<p>
We now offer overnight delivery.
<h2>Everything your pet needs</h2>
```

Notice the three sizes of text used in the example: <h1> for large text, <h2> for almost-as-large text, and body text. If Arlo forgot to use the </h1> tag at the end of the first line, the body text would appear huge, too. The second line includes the <p> tag, the end-of-paragraph mark. This tells the Web browser to end the line and skip some space before the next line. HTML normally ignores "white space" (returns, spaces and tabs in your file), so the code:

```
Make
yourself
at
home.
```

appears as "Make yourself at home." Why are there no <p> tags after the </h1> and before the <h2>? You've just discovered one of HTML's idiosyncrasies: you don't need to use a paragraph mark right before a headline size change. (You can include it, but things look better if you don't. Try it both ways.)

There are many ways to tweak text, including tags for **bold text** and *italic text* and <u>underlined text</u> and a <tt>monospace "typewriter" font</tt>. Those are all "physical styles"—HTML allows you to specify text appearance in two ways: physically (by explicitly specifying the style you

©1995 Thompson Publishing Group

want, such as italics, boldface, or underline) and logically (by the logical meaning of the text you wish to mark), such as emphasis and strong emphasis. It's considered better to use logical style tags rather than physical ones, because logical tags are more consistently interpreted from browser to browser.

You'll notice that you don't have a lot of control over how text appears. If you're accustomed to using a word processor in which you can choose from umpteen fonts, styles, and point sizes, HTML can seem pretty limiting. HTML was designed to work across many platforms, to give the reader, rather than the publisher, control over how things appear. This is slowly changing, but for now the control an HTML publisher can exert is limited.

Lists

HTML offers a variety of tools for presenting lists of information, including numbered lists, unnumbered lists, and "glossary" lists. For instance,

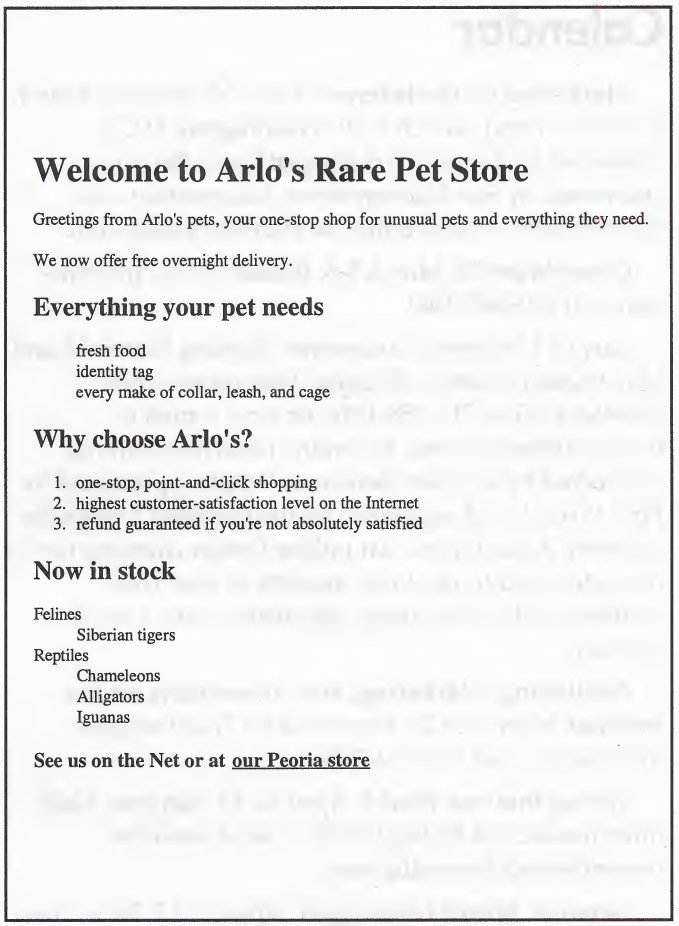
```
<h2>Everything your pet needs</h2>
<ul>
<li>fresh food
<li>identity tag
<li>every make of collar, leash, and cages
</ul>
```

will create an *unordered* list, using bullets—or nothing, depending on browser—instead of numbers to separate the items. By changing the UL tags to OL tags, we can make an *ordered*, or numbered, list. Notice that the browser automatically numbers the items, making it painless to add, move or remove list items. Some browsers also let you specify the type of number or ordering device displayed, such as Roman numerals (I, II, III) or letters (A, B, C).

```
<h2>Why choose Arlo's?</h2>
<ol>
<li>one-stop, point-and-click shopping
<li>highest customer-satisfaction level on
the Internet
<li>refund guaranteed if you're not abso-
lutely satisfied
</ol>
```

Another type of list, the glossary, or *definition list* (dl), is useful for creating items with subelements. For instance, the HTML code:

```
<h2>Now in stock</h2>
<dl>
<dt>Felines
<dd>Siberian tigers
[etc.]
</dl>
```



Welcome to Arlo's Rare Pet Store

Greetings from Arlo's pets, your one-stop shop for unusual pets and everything they need.

We now offer free overnight delivery.

Everything your pet needs

fresh food
identity tag
every make of collar, leash, and cage

Why choose Arlo's?

1. one-stop, point-and-click shopping
2. highest customer-satisfaction level on the Internet
3. refund guaranteed if you're not absolutely satisfied

Now in stock

Felines
Siberian tigers
Reptiles
Chameleons
Alligators
Iguanas

See us on the Net or at our Peoria store

Images

Of course, you can also place images in your Web page. The IMG tag allows an image to be inserted within a document. The easiest way to include an image is to put a GIF-format graphic in the same directory as your HTML code, then use a line like: <img align=bottom src="filename.gif">. This tells the Web browser to place an image called filename.gif on the page, and make the next text or picture align with the bottom of the picture.

Using the tag <img align=top src="arlodad.gif"> aligns the image with the preceding element.

In addition to these "inline graphics," you can also make available graphics that a user can download as separate files by clicking on a highlighted link.

Linking is at the heart of HTML. The figure shows a sample link created with the following code:

```
<a href="http://www.arlo.com/
store.html">our Peoria store</a>
```

The next article in this series covers linking to other Web pages, and introduces you to selected, more advanced, aspects of HTML.

©1995 Thompson Publishing Group